Aho Ullman

- Instruction selection
- over optimized IR using cost based tree pattern matching

Davidson Fraser

- Instruction selection
- over AST using structural tree pattern matching naive code which is ▷ target dependent, and is ▷ optimized subsequently
Configuring GCC

Notes

Bootstrapping: The Conventional View

Notes
A Native Build on i386

Requirements:
- BS = HS = TS = i386
- Stage 1 build compiled using cc
- Stage 2 build compiled using gcc
- Stage 3 build compiled using gcc
- Stage 2 and Stage 3 Builds must be identical for a successful native build

Build for a Given Machine

This is what actually happens!

- Generation
  - Generator sources
    - $(SOURCE_D)/gcc/gen*.c are read and generator executables are created in $(BUILD)/gcc/build
    - MD files are read by the generator executables and back end source code is generated in $(BUILD)/gcc

- Compilation
  - Other source files are read from $(SOURCE_D) and executables created in corresponding subdirectories of $(BUILD)

- Installation
  - Created executables and libraries are copied in $(INSTALL)
**More Details of an Actual Stage 1 Build for C**

- Libraries
- Libiberty
- Fixincl
- Gen*
- Cc
- Cpp
- Xgcc
- Libgcc
- Target binutils
- Cc + binutils for stage 2

**Basic Transformations in GCC**

Transformation from a language to a *different* language

- Target Independent
- Parse ➔ Simplify ➔ Tree SSA
- Optimize
- Generate RTL ➔ Optimize RTL ➔ Generate ASM
- GIMPLE ➔ RTL
- RTL ➔ ASM

- Target Dependent

**Notes**
Instruction Specification and Translation: A Recap

- Target Independent
- Target Dependent

- GIMPLE: target independent
- RTL: target dependent
- Need: associate the semantics

⇒ GCC Solution: Standard Pattern Names

- GIMPLE
  - ASSIGN
  - ⇒ RTL Template

(define_insn "movsi"
  [(set (match_operand 0 "register_operand" "r")
    (match_operand 1 "const_int_operand" "k"))]
  "/* C boolean expression, if required */
  "li %0, %1"
)

Notes

Translation Sequence in GCC

(define_insn
  "movsi"
  [(set
    (match_operand 0 "register_operand" "r")
    (match_operand 1 "const_int_operand" "k"))]
  "/* C boolean expression, if required */
  "li %0, %1"
)

D.1283 = 10;
(set
  (reg:SI 58 [D.1283])
  (const_int 10: [0xa])
)⇒ li $t0, 10

Notes
Retargetability Mechanism of GCC

Plugin Structure in `cc1`
The GNU Tool Chain for LTO Support

```
cc1'  lto1'  cc1

common

"Fat" .s files
as as

"Fat" .o files
cc1'  lto1'  lto1
common

Single .s file
as as

Single .o file + glibc/newlib
lto1
collect2

a.out file
```

```
Common Code (executed twice for each function in the input program)
cgraph_optimize
  ipa_passes
    execute_ipa_pass_list(all_small_ipa_passes)/*!in lto*/
    execute_ipa_summary_passes(all_regular_ipa_passes)
    execute_ipa_summary_passes(all_lto_gen_passes)
    ipa_write_summaries
cgraph_expand_all_functions
cgraph_expand_function
  /* Intraprocedural passes on GIMPLE, */
  /* expansion pass, and passes on RTL. */
```
Hooking up Back End Details

Runtime initialization of data structure

Essential Abstractions in GCC
GCC Resource Center, IIT Bombay